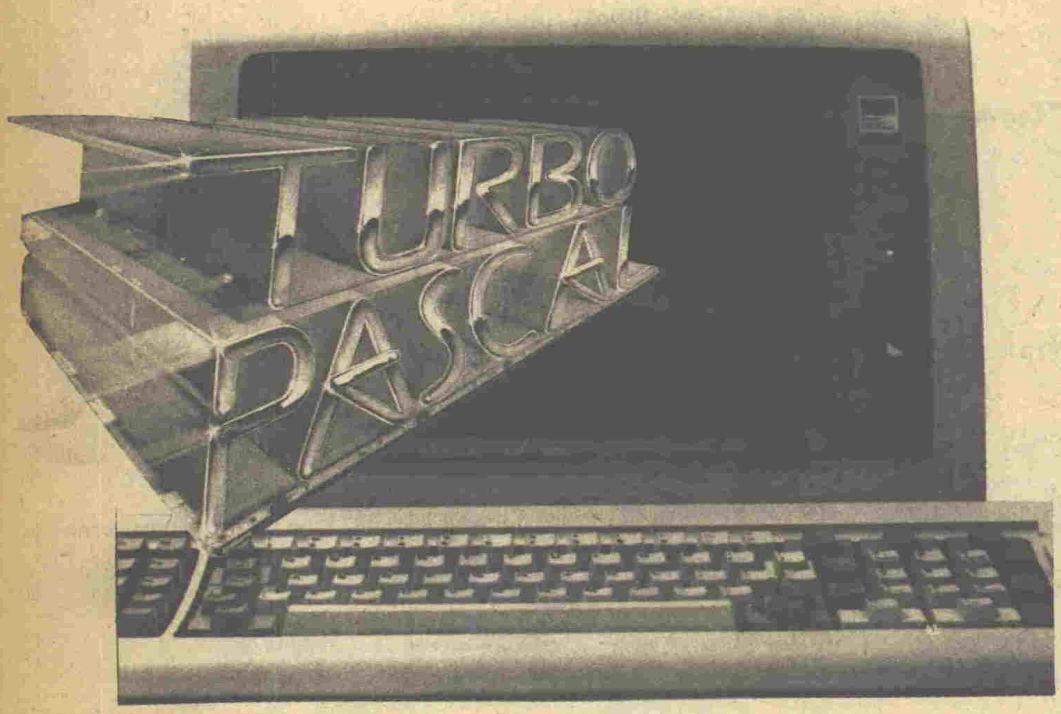


กราฟิค บน เทอร์โบปาสคาลเวอร์ชั่น 3.0

วิชัย ประเสริฐเจริญสุข

ท่านคงทราบดีว่าเทอร์โบปาสคาลเป็นภาษาที่ดีมากภาษาหนึ่งและนิยมใช้กันมาก ปัจจุบันภาษานี้ได้พัฒนาขึ้นมาจนกระทั่งสามารถที่จะนำไปเขียนรูปต่าง ๆ ได้ดีมาก ที่จะกล่าวต่อไปนี้คือ เทอร์โบปาสคาลเวอร์ชั่น 3.0 ซึ่งจะทำงานบนเครื่องไอบีเอ็มพีซี ในความคิดของผมเองผมคิดว่าภาษานี้สามารถเขียนรูปต่างๆ ได้ดีกว่าภาษาอื่นๆ ทั้งนี้เพราะว่ามันมีคำสั่งในการเขียนรูปต่าง ๆ มากมายและใช้ได้สะดวกอีกด้วย ตัวอย่างเช่น การทำช่องหน้าต่าง การเขียนวงกลม การเขียนส่วนโค้งของวงกลม การเก็บภาพบางส่วนบนจอภาพแล้วนำไปเขียนอีกทีหนึ่งได้ การระบายสีภายในขอบเขต การเขียน pattern โดยใช้คำสั่งเพียงคำสั่งเดียว และยังสามารถเขียนรูปต่าง ๆ เหมือนกับคำสั่งในภาษาโลโกได้อีกด้วย

การแสดงเอาท์พุทบนจอภาพสามารถแสดงได้ 2 โหมดใหญ่ ๆ คือ

1. textmode สามารถแบ่งออกเป็น 80, 40 ตัวอักษรและยังให้แสดงเป็นตัวอักษรสีหรือขาวดำ (ใช้ BW หมายถึง ขาวดำ C หมายถึง สี) ตัวอย่างเช่น

textmode (BW40) หมายถึงให้แสดงตัวอักษรแบบขาวดำ 40 ตัวอักษรต่อแถว

textmode (C80) หมายถึงให้แสด
ตัวอักษรแบบสี 80 ตัวอักษรต่อแถว

2. graphmode สามารถแบ่งโหมดการทำงา
ได้เป็น 3 โหมด คือ

graphcolormode; เป็นโหมดที่ใ
เขียนรูปสีด้วยขนาด 320×200 จุด

graphmode; เป็นโหมดที่ใช้เขียนรู
ขาวดำด้วยขนาด 320×200 จุด

hires; เป็นโหมดเขียนภาพสีและ
ความละเอียดสูง คือ 640×200 จุด

ก่อนที่เราจะดูคำสั่งในการเขียนรู
โดยใช้เทอร์โบปาสคาล ผมขอกล่าวถึงกา
ทำช่องหน้าต่างบน text เสียก่อนเพราะว
เราสามารถที่จะเขียนตัวอักษรลงในกรา
โหมดได้เลยโดยใช้คำสั่ง write เข้าไปเท่านั้
ลองดูตัวอย่างในรูปที่ 1 ซึ่งเป็นโปรแกร
การทำช่องหน้าต่าง ให้ชื่อว่า test 1 โ
จะให้ตัวอักษรที่ต้องการเขียน เขียนลงภาย
ขอบเขตสี่เหลี่ยมที่กำหนดไว้เท่านั้น ก่อนอื่
เรากำหนดให้ text ของเราอยู่ในโหมดตั
อักษรเป็นสีและมีขนาด 40 ตัวอักษรต่อ
แถว ต่อมาเราก็กำหนดให้พื้นของจอภา
เป็นสี 0 หรือสีดำเราจะเลือกสีอื่นได้ตา
ตารางในรูปที่ 2 แล้วกำหนดให้ตัวอักษร
จะแสดงเป็นสีเหลือง อาจจะใช้สีอื่นก็
ใช้ procedure write boundary เป็นก
เขียน * เพื่อแสดงให้เห็นชัดถึงขอบเขตข
ช่องหน้าต่าง เมื่อเรากำหนด window (x
y_1, x_2, y_2) หมายถึงขอบเขตของช่องหน้าต่
ให้อยู่ในตำแหน่ง 10, 10 ถึงจุด 26,
อย่าสลับที่กันเป็นอันขาด จะต้องกำห
จุดบนซ้ายมือสุดก่อนแล้วจึงกำหนดจุดล่
ขวามือ เมื่อกำหนดช่องหน้าต่างแล้วจะต้
มีคำสั่งให้เคอร์เซอร์มาอยู่ภายในช่องหน้าต่
คือคำสั่ง gotoxy โปรแกรม test 1 นี้ให้เขี
ตัวอักษรโดยการเลือกตัวอักษรแบบสุ่มตัวเ
เมื่อจะจบโปรแกรมจะต้องมีการสั่งทำช่
หน้าต่างใหม่ให้เป็นปกติ คือ window (
1, 40, 25) ไม่เช่นนั้นเวลารันใหม่ คำ
window เดิมจะยังคงอยู่ โปรแกรมนี้จะเขี
ตัวอักษรไปเรื่อย ๆ จนกระทั่งเรากดคีย์
คีย์หนึ่งจึงจะออกจากโปรแกรม

```
program  test1;
  var  i:integer;

procedure  writeboundary;
   begin
      gotoxy(8,8);write('**********************');
      gotoxy(8,23);write('**********************');
      for i:=9 to 23 do
        begin
           gotoxy(8,i);write('*');
           gotoxy(28,i);write('*');
        end;
   end;
begin
      textmode(c40);
      textbackground(0);
      textcolor(yellow);
      gotoxy(8,4);
      writeln(' THIS OUT OF WINDOW  ');
      writeboundary;
      gotoxy(10,10);
      window(10,10,26,21);
      repeat
        i:=random(94)+33;
        write(chr(i));delay(20);
      until keypressed;
      window(1,1,40,25);
      textmode;
end.
```

รูปที่ 1 โปรแกรม test 1 สำหรับทำช่องหน้าต่าง

คำสั่งพื้นฐานในโหมดกราฟิค

คำสั่งที่นอกเหนือจากคำสั่งเลือกโหมดที่กล่าวมาแล้วยังมีคำสั่งพื้นฐานอีกดังนี้

– graphbackground(color) เป็นคำสั่งเลือกสีพื้นบนจอภาพซึ่งมีค่า 0 ถึง 15 ตามรูปที่ 2

palette(N); เป็นคำสั่งเลือกแถบสีใช้ได้ใน graphcolormode ซึ่งมีค่าตามตารางรูปที่ 3

hirescolor(color); เป็นคำสั่งเลือกสีที่จะทำการเขียนรูปมีค่าตามรูปที่ 2 ใช้ใน hiresmode

plot(x,y,color); เป็นคำสั่งให้เขียนเป็นจุดที่ตำแหน่ง x,y โดยใช้สีตาม color ซึ่งมีค่า 0 ถึง 3 ใน graphcolormode และ 0 ถึง 1 ใน hiresmode

draw (x_1, y_1, x_2, y_2, color); เป็นคำสั่งให้ลากเส้นจากจุด x_1, y_1 ไปยังจุด x_2, y_2

graphwindow (x_1, y_1, x_2, y_2); เป็นคำสั่งสร้างช่องหน้าต่างในกราฟิคโหมดทุกโหมด ซึ่งค่า x_1, y_1 จะต้องเป็นจุดซ้ายมือด้านบนของจอภาพและจุด x_2, y_2 จะต้องเป็นจุดขวามือด้านล่างของจอภาพเสมอ

Dark Color	Light Color
0: black	8: drakgray
1: blue	9: lightblue
2: green	10: lighgreen
3: cyan	11: lightcyan
4: red	12: lightred
5: magenta	13: lightmagenta
6: brown	14: yellow
7: lightgray	15: white

รูปที่ 2 ตารางรหัสสี

Color number	0	1	2	3
Palette 0	background	green	red	brown
Palette 1	background	cyan	magenta	lightgray
Palette 2	background	lightgreen	lightred	yellow
Palette 3	background	lightcyan	lightmagenta	white

รูปที่ 3 ตารางแสดงแถบสี

คำสั่งเพิ่มเติมในโหมดกราฟิค

ที่กล่าวมาแล้วเป็นคำสั่งพื้นฐานที่สามารถเขียนรูปได้เลย แต่ต่อไปนี้เป็นคำสั่งที่เพิ่มขึ้นมาจากเดิม ซึ่งจะนำไปใช้ประโยชน์ได้มาก ก่อนที่เราจะใช้คำสั่งเหล่านี้ เราจะต้องมีไฟล์ 2 ไฟล์อยู่ในดิสค์แล้ว ซึ่งเป็นไฟล์สำเร็จรูปคือ graph.P และ graph.BIN ดังนั้นก่อนที่จะโปรแกรมคำสั่งเหล่านี้จะต้องเรียก incluse file graph.P ก่อนทุกครั้งดังนี้

{ $I graph.P }

– colortable (c_1, c_2, c_3, c_4); เป็นคำสั่งสร้างตารางสีของการเขียนรูป แต่ละครั้งใช้ได้ใน graphiccolormode ก่อนที่เราจะกำหนดตารางสีเราจะต้องกำหนดแถบสี (palette) เสียก่อน ปกติเครื่องจะกำหนดตารางสีไว้ก่อนแล้วคือเป็น colortable (0, 1, 2, 3) ถ้าเราใช้คำสั่งว่า colortable (1, 2, 3, 0) แล้วลากเส้นด้วยคำสั่งว่า draw (x_1, y_1, x_2, y_2, -1) จะหมายความว่าก่อนที่จะลากเส้นจะต้องดูพื้นที่มันจะลากเส้นก่อนว่าเป็นสีอะไร ถ้าเป็นสี 0 มันจะลากเส้นด้วยสี 1 ถ้าพื้นเป็นสี 1 มันจะลากเส้นเป็นสี 2 ถ้าพื้นเป็นสี 2 มันจะลากเส้นเป็นสี 3 เป็นต้น และทุกครั้งที่ต้องการใช้ colortable คำสั่งในการเขียนรูป color จะต้องมีค่าเท่ากับ –1 เสมอ

– getdotcolor (x, y); เป็นคำสั่ง

ทดสอบจุดบนจอภาพว่าเป็นจุดสีอะไร ซึ่งจะเป็นตัวเลขจำนวนเต็ม ถ้าอยู่ใน graphcolormode จะมีค่าเท่ากับ 0 ถึง 3 ถ้าอยู่ใน hires จะมีค่า 0 และ 1 ถ้าจุดอยู่นอกช่องหน้าต่างจะมีค่าเท่ากับ −1 ดังโปรแกรม test 2 ตามรูปที่ 4

โปรแกรม test 2 นี้เป็นโปรแกรมสำหรับทดลองใช้คำสั่ง colortable เพื่อให้เข้าใจได้ง่ายขึ้น เราจะเห็นการเปลี่ยนแปลงได้ชัดเจนเมื่อเรามีจอภาพเป็นสีแต่ถ้าไม่มีจอภาพเป็นสี ก็สามารถดูได้จากตัวเลขหลัง testcolor ซึ่งจะเป็นตัวเลข 0 ถึง 3 เป็นตัวแสดงสีที่จุด 100, 140 และเป็นที่สังเกตเราสามารถเขียนตัวอักษรลงบนกราฟิคได้ ซึ่งจะมีขนาด 25×40 แต่ถ้าเป็น hires จะมีขนาด 25×80

– arc (x, y, angle, radius, color); เป็นคำสั่งในการเขียนรูปส่วนโค้งของวงกลม ซึ่งมีจุดศูนย์กลางที่จุด x, y angle เป็นตัวบอกมุมเป็นองศามีทั้งค่าลบและบวก ถ้าเป็นค่าบวกจะนับตามเข็มนาฬิกา ส่วนค่าลบจะนับทวนเข็มนาฬิกา radius เป็นค่าของรัศมี color เป็นค่าสี ถ้าเป็น −1 จะเป็นไปตาม colortable

– circle (x, y, radius, color); เป็นคำสั่งในการเขียนวงกลม ส่วนค่าต่าง ๆ เหมือนกับคำสั่ง arc

– getpic (buffer, x_1, y_1, x_2, y_2); เป็นคำสั่งเก็บภาพที่ปรากฏบนจอภาพ มีขอบเขตเป็นรูปสี่เหลี่ยม ที่กำหนดโดย x_1, y_1 และ x_2, y_2 เก็บไว้ที่ตัวแปร buffer ซึ่งจะเป็นตัวแปรชนิดใดก็ได้ แต่จะต้องมีขนาดของไบท์มากพอที่จะเก็บดาต้าของภาพไว้ได้ จำนวนไบท์ต่ำสุดในการเก็บภาพคือ

320 × 200 modes:

size = ((width + 3) div 4) · height + 6

640 × 200 mode:

size = ((width + 7) div 8) · height + 6

เมื่อ width = abs (x_1−x_2) + 1

และ height = abs (y_1−y_2) + 1

– putpic (buffer, x, y); เป็นคำสั่งให้เขียนรูปที่เก็บไว้โดยคำสั่ง getpic ค่าของ x, y เป็นตำแหน่งที่อยู่ซ้ายมือด้านล่างของภาพ

– fillscreen (color); เป็นคำสั่งให้ระบายสีภายในช่องหน้าต่างที่กำหนดไว้ color มีค่า 0 ถึง 3 ถ้ามีค่า −1 จะทำงานตาม colortable

– fillshape (x, y, fillcolor, bordercolor); เป็นคำสั่งให้ระบายสีภายในขอบเขตที่กำหนดไว้ โดย fillcolor คือสีที่ต้องการระบาย จุด x, y เป็นจุดที่อยู่ภายในบริเวณที่ต้องการระบายสี bordercolor เป็นขอบเขตของสีที่ต้องการให้ระบายภายใน

– pattern (P); เป็นคำสั่งในการเขียนรูปขนาด 8×8 จุด P เป็นพารามิเตอร์ที่มี array[0..7]of byte ซึ่งแต่ละไบท์หมายถึงจุดตามแนวนอน ถ้าบิทใดเป็น 1 หมายถึงให้เขียนจุดสีตามคำสั่ง fillpattern ถ้าบิทใดเป็น 0 ก็ไม่ต้องเขียนจุด ลองดูการเขียน pattern ดังรูปที่ 5

จากรูปที่ 5 เราสามารถเขียน pattern ได้ดังนี้

```
const
Lines : array | 0..7 | of byte =
        ($44, $88, $11, $22,
         $44, $88, $11, $22) ;
```

– fillpattern (x_1, y_1, x_2, y_2, color); เป็นคำสั่งให้ใส่ pattern ที่เราได้จัดทำไว้ก่อนแล้วลงในบล็อกที่มีจุดบอกคือ x_1, y_1 และจุด x_2, y_2 และ color เป็นสีที่เราต้องการให้เขียน

โปรแกรม test 3 ตามรูปที่ 6 นี้เป็นโปรแกรมที่ทดลองใช้คำสั่งต่าง ๆ ที่กล่าวมาแล้ว รูปที่เราได้จะเป็นไปตามรูปที่ 7 ครั้งแรกเมื่อรันโปรแกรมเราจะได้ภาพของวงกลม (สี 2 หรือสีแดง) ขึ้นมา 3 ลูกตัดกัน ถ้าเรากดคีย์ใดคีย์หนึ่ง เราจะเห็นบล็อกเพิ่มขึ้นมาหนึ่งบล็อกซึ่งเกิดจาก pattern ที่เรากำหนด

```
program  test2;
  {$i graph.p}
  var ch:char;i,n:integer;
  begin
     graphcolormode;
     graphbackground(black);
     palette(0);
     textcolor(1);
     writeln('                    table ');
     writeln('0 : black; 1 : green; 2 : red; 3 : brown');

     gotoxy(1,5);write('  test color (100,140) :');
     colortable(1,2,3,0);
     n:=50;
     repeat
     for i:= n to 150 do
          draw(10,i,290,i,-1);

     n:=n+25;
     write('  ',getdotcolor(100,140));
     read(kbd,ch);
     until n=150;

     textmode;
  end.
```

รูปที่ 4 โปรแกรม test 2 ใช้กำหนดสี

```
– * --- * -- $ 44
 * --- * --- $ 88
--- * --- * $ 11
-- * --- * – $ 22
– * --- * -- $ 44
 * --- * --- $ 88
--- * --- * $ 11
-- * --- * – $ 22
```

รูปที่ 5 ตัวอย่างการเขียน pattern

ไว้ และถ้ากดคีย์อีกมันจะระบายสี 1 หรือสีเหลือง ภายในขอบของวงกลมสีแดง ถ้าเรากดคีย์อีกมันจะเก็บภาพที่มีครึ่งวงกลมและบล็อกรวมอยู่ด้วย แล้วนำเอาภาพนั้นไปเขียนลงในภาพด้านขวามือ 3 รูป ภาพนี้จะมีขนาด width = 51, height = 61 เพราะฉะนั้นขนาดของไบท์ต่ำสุดของภาพนี้คือ 793 ไบท์ เมื่อกดคีย์อีกครั้งหนึ่งจะเห็นดาต้าที่เก็บใน interger variable ของ picture ซึ่งจะมีขนาด 397 ตัว ที่ไม่เป็น 0 แสดงให้เห็นว่ารูปภาพที่เก็บไว้ในบัฟเฟอร์ไม่เกิน 793 ไบท์ รูปภาพนี้จะเก็บไว้ใช้อีกได้ตลอดไป ถ้าเราเก็บเข้าเป็นไฟล์

กราฟิคของเต่า (turtle graphics)

เทอร์โบปาสคาลนอกจากมีคำสั่งในการเขียนรูปต่าง ๆ ที่กล่าวมาแล้ว ยังมีลักษณะเด่นอีกประการหนึ่งคือ สามารถเขียนรูปในลักษณะคล้ายกับภาษาโลโกได้อีกด้วย การทำงานลักษณะนี้เป็นการทำงานโดยอาศัยการเคลื่อนที่ของเต่า ซึ่งมีรูปร่างเป็นรูปสามเหลี่ยมให้เคลื่อนที่ไปในทิศทางที่ต้องการพร้อมกับลากเส้นหรือไม่ลากเส้นก็ได้ การทำงานของเต่าในโหมดกราฟิคทุกครั้ง จะต้องเรียก incluse file graph.P ก่อนทุกครั้ง

ขอบเขตของตำแหน่งของเต่า

320 x 200 modes :
x = -159..0.. 160,
y = -99..0..100
640 x 200 modes :
x = -319..0..320,
y = -99..0..100

คำสั่งของเต่าในโหมดกราฟิค

– back (dist); หรือ forwd (dist); เป็นคำสั่งให้เต่าเคลื่อนที่ถอยหลังหรือเดินหน้าด้วยระยะทาง dist

– pen down; หรือ pen up; เป็นคำสั่งให้มีการเขียนเส้นตรงด้วยหรือไม่ในกรณีที่มีการเคลื่อนที่

– set pen color (color); เป็นคำสั่งกำหนดสีที่จะเขียนรูปซึ่งมีค่า –1 ถึง 3

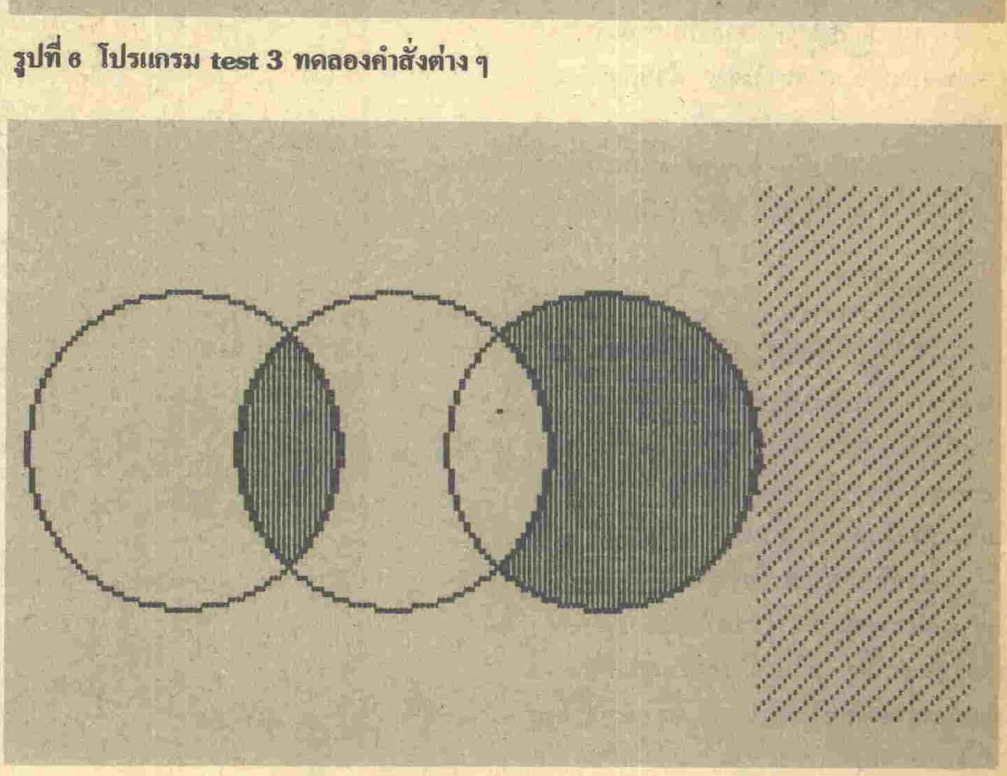
– show turtle; hide turtle; เป็นคำสั่งให้แสดงตัวเต่าหรือไม่

– home; เป็นคำสั่งให้เต่าอยู่ในตำแหน่ง 0, 0

– heading; เป็นคำสั่งให้เต่าหันหัวไปทิศเหนือหรือด้านบนของจอภาพ

```
program  test3;
  const
       dot: array[0..7] of byte =
            ($44,$88,$11,$22,$44,$88,$11,$22);
  var  ch:char;i:integer;
  var  picture:array[1..410] of integer;
  {$i graph.p}
  begin
       graphcolormode;
       palette(0);
       circle(80,100,30,2);
       circle(120,100,30,2);
       circle(160,100,30,2);
       pattern(dot);
       read(kbd,ch);
       fillpattern(190,150,230,50,1);
       read(kbd,ch);
       fillshape(100,100,1,2);
       fillshape(160,100,1,2);
       read(kbd,ch);
       getpic(picture,160,130,210,70);
       putpic(picture,240,130);delay(1000);
       putpic(picture,240,60);delay(1000);
       putpic(picture,240,200);
       read(kbd,ch);
       textmode(bw80);
       for i:= 1 to 410 do
           writeln(i,' ',picture[i]);
  end.
```

รูปที่ 6 โปรแกรม test 3 ทดลองคำสั่งต่าง ๆ



รูปที่ 7 ผลของโปรแกรม test 3

- set heading (angle); เป็นคำสั่งกำหนดทิศทางของตัวเต่า มีค่ามุม 0..395

- turn left (angle), หรือ turn right (angle); เป็นคำสั่งหมุนทิศทางของตัวเต่าจากเดิมไปกี่องศา ซึ่งมีค่า 0..359 ค่านี้อาจจะเป็นลบก็ได้หมายถึงให้หมุนกลับทาง

- set position (x, y); เป็นคำสั่งกำหนดตำแหน่งของเต่า

- turtle window (x, y, w, h); เป็นคำสั่งในการสร้างช่องหน้าต่างซึ่งค่า x, y เป็นตำแหน่งของ graphmode, w คือความกว้างตามแนวแกน x, h คือความสูงตามแนวแกน y

- clear screen; เป็นคำสั่งลบจอภาพภายในช่องหน้าต่าง

- wrap; และ no wrap; เป็นคำสั่งให้มีการเคลื่อนเต่า ในด้านตรงข้ามกับทิศทางการเคลื่อนที่ของเต่าหรือไม่ในกรณีที่มีการเคลื่อนที่ออกนอกช่องหน้าต่าง

- turtle there; เป็นบูลลีน ถ้า true หมายถึงเต่า อยู่ภายในช่องหน้าต่าง ถ้า false หมายถึงอยู่นอกช่องหน้าต่าง

- Xcor; Ycor; เป็นตัวเลขจำนวนเต็มที่บอกตำแหน่งของเต่าทางแกน x และ y

- turtle delay(MS); เป็นคำสั่งให้การทำงานของเต่า แต่ละขั้นช้าลงด้วยเวลา MS (msec)

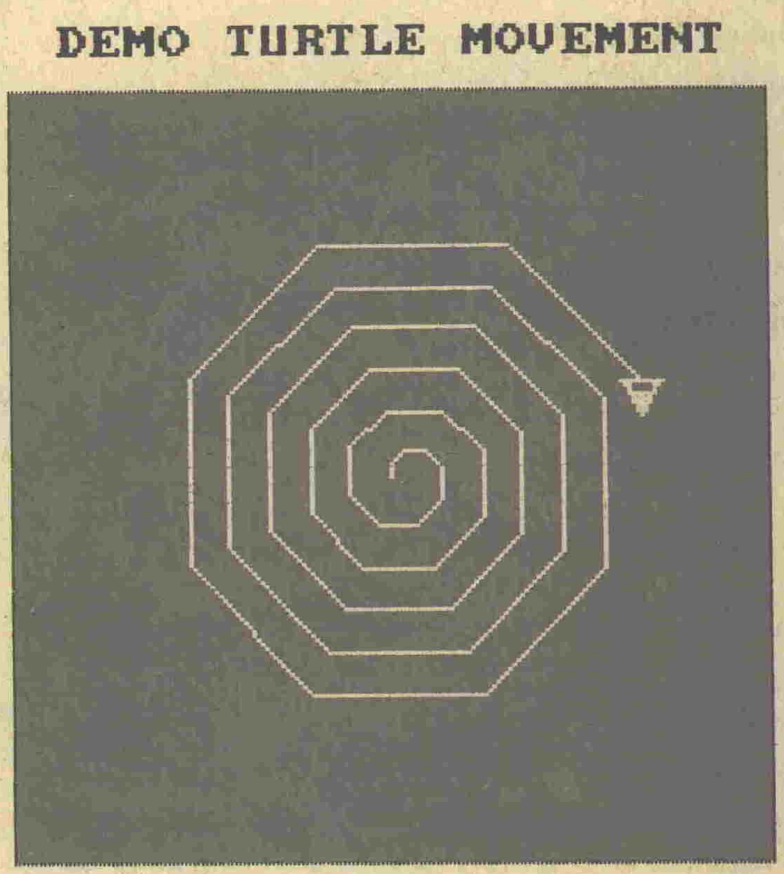
- sound(I); nosound; เป็นคำสั่งให้ลำโพงดังและหยุดด้วยความถี่ I มีหน่วยเป็นเฮิรทซ์

จากโปรแกรม test 4 ตามรูปที่ 8 นี้จะเห็นได้ว่าเราให้เต่าเคลื่อนที่วนตามเข็มนาฬิกาในลักษณะหักมุม 45 องศาและระยะทางเพิ่มขึ้นทีละหนึ่ง ภายในช่องหน้าต่างที่กำหนดไว้ดังรูปที่ 9 การเคลื่อนที่ทุกครั้งจะมีเสียงดังด้วยความถี่หนึ่ง

จากที่กล่าวมาแล้วในคำสั่งต่าง ๆ ของกราฟิคคงพอจะบอกได้ว่าเทอร์โบปาสคาล–เวอร์ชั่น 3 นี้คงจะมีประโยชน์กับเราเป็นอย่างมากในการเขียนรูปต่าง ๆ พร้อมทั้งภาษาปาสคาลยังเป็นภาษาที่เหมาะมากในการเล่นกราฟิค

```
program test4;
   var  i:integer;
        ch:char;
   {$i graph.p}
   procedure  drawwindow;
      begin
         draw(69,20,249,20,1);
         draw(249,20,249,199,1);
         draw(249,199,69,199,1);
         draw(69,199,69,20,1);
      end;
   begin
      graphcolormode;
      palette(0);textcolor(3);
      gotoxy(10,1);writeln(' DEMO TURTLE MOVEMENT');
      drawwindow;
      turtlewindow(159,110,180,180);
      showturtle;
      home;setpencolor(2);
      i:=3;read(kbd,ch);
      repeat
        forwd(i);sound(400);delay(300);
        turnright(45);sound(800);delay(300);
        i:=i+1;
        nosound;
      until keypressed;
      textmode(bw80);
   end.
```

รูปที่ 8 โปรแกรม test 4 แสดงการเคลื่อนที่ของเต่า



รูปที่ 9 แสดงการเคลื่อนที่ของเต่า